

Correction du DS 1

Informatique de tronc commun, première année

Julien REICHERT

Exercice 1

```
def trouve_minimum(l):
    rep = 0
    mini = l[0]
    for i in range(1, len(l)):
        if l[i] < mini:
            rep = i
            mini = l[i]
    return rep
```

Exercice 2

```
def taille_totale(L):
    rep = 0
    for sousliste in L:
        rep += len(sousliste)
    return rep
```

Exercice 3

```
def maxitotal(L):
    rep = None # précaution si L[0] est vide
    for sousliste in L:
        for element in sousliste:
            if rep == None or rep < element:
                rep = element
    return rep
```

Exercice 4

```
def tous_couples(l, ll):
    rep = []
    for element in l:
        for ellement in ll:
            rep.append((element, ellement)) # attention à bien mettre deux parenthèses
    return rep
```

Exercice 5

```
def inclus(l, ll): # sous-fonction pour pouvoir utiliser return, plus simple
    for element in l:
        if element not in ll:
            return False
    return True

def une_inclusion(l, ll):
    return inclus(l, ll) or inclus(ll, l)
```

Exercice 6

Il s'agit d'une fonction qui construit toutes les sous-listes de la liste en argument, en maintenant l'ordre des éléments. Une liste de taille n produira donc une liste de taille 2^n .

Exercice 7

Ici, la boucle `for` est perturbée par la mutation de la liste en cours de parcours. Il s'agira alors en pratique d'une boucle infinie.